

UC Irvine

UC Irvine Previously Published Works

Title

Deep Neural Network Cloud-Type Classification (DeepCTC) model and its application in evaluating PERSIANN-CCS

Permalink

<https://escholarship.org/uc/item/30v9742x>

Journal

Remote Sensing, 12(2)

ISSN

2072-4292

Authors

Gorooh, VA
Kalia, S
Nguyen, P
[et al.](#)

Publication Date

2020

DOI

10.3390/rs12020316

Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at <https://creativecommons.org/licenses/by/4.0/>

Peer reviewed



A Deep Neural Network Modeling Framework to Reduce Bias in Satellite Precipitation Products

YUMENG TAO, XIAOGANG GAO, KUOLIN HSU, AND SOROOSH SOROOSHIAN

Department of Civil and Environmental Engineering, University of California, Irvine, Irvine, California

ALEXANDER IHLER

Department of Computer Science, University of California, Irvine, Irvine, California

(Manuscript received 6 May 2015, in final form 28 December 2015)

ABSTRACT

Despite the advantage of global coverage at high spatiotemporal resolutions, satellite remotely sensed precipitation estimates still suffer from insufficient accuracy that needs to be improved for weather, climate, and hydrologic applications. This paper presents a framework of a deep neural network (DNN) that improves the accuracy of satellite precipitation products, focusing on reducing the bias and false alarms. The state-of-the-art deep learning techniques developed in the area of machine learning specialize in extracting structural information from a massive amount of image data, which fits nicely into the task of retrieving precipitation data from satellite cloud images. Stacked denoising autoencoder (SDAE), a widely used DNN, is applied to perform bias correction of satellite precipitation products. A case study is conducted on the Precipitation Estimation from Remotely Sensed Information Using Artificial Neural Networks Cloud Classification System (PERSIANN-CCS) with spatial resolution of $0.08^\circ \times 0.08^\circ$ over the central United States, where SDAE is used to process satellite cloud imagery to extract information over a window of 15×15 pixels. In the study, the summer of 2012 (June–August) and the winter of 2012/13 (December–February) serve as the training periods, while the same seasons of the following year (summer of 2013 and winter of 2013/14) are used for validation purposes. To demonstrate the effectiveness of the methodology outside the study area, three more regions are selected for additional validation. Significant improvements are achieved in both rain/no-rain (R/NR) detection and precipitation rate quantification: the results make 33% and 43% corrections on false alarm pixels and 98% and 78% bias reductions in precipitation rates over the validation periods of the summer and winter seasons, respectively.

1. Introduction

Weather forecasts, climate variability, hydrology, and water resources management require sufficient information about precipitation, one of the most important variables in the natural water cycle. Precipitation observation, monitoring, and analysis tools provide fundamental information needed in order for society to cope with increasing extreme hydrometeorological events in recent decades. Satellite-based precipitation products mainly estimate precipitation indirectly based on information collected from multiple wavelengths. Common

choices include visible (VIS) and infrared (IR) wavelength images of cloud albedo and cloud-top brightness temperature from geosynchronous satellites (Hsu et al. 1997; Nasrollahi et al. 2013). Another popular data source is passive microwave (PMW) images from sensors on board low-Earth-orbiting (LEO) satellites. PMW images provide information about the atmospheric constituents and hydrometeorological profiles, which is more directly related to the ground precipitation rate (Joyce et al. 2004; Kidd et al. 2003). One undesirable part of PMW is its relatively low temporal resolutions (Marzano et al. 2004).

Several operational satellite precipitation products are available for public use through their open websites. The Climate Prediction Center (CPC) morphing technique (CMORPH), developed by the National Oceanic and Atmospheric Administration (NOAA), uses precipitation

Corresponding author address: Yumeng Tao, Department of Civil and Environmental Engineering, University of California, Irvine, E4130 Engineering Gateway, Irvine, CA 92697.
E-mail: yumengt@uci.edu

TABLE 1. Input features extracted for PERSIANN-CCS.

Coldness features of cloud patch	Min temperature of a cloud patch Mean temperature of a cloud patch
Geometric features	Cloud-patch area Cloud-patch shape index
Texture features	Std dev of cloud-patch temperature Mean value of local std dev of cloud temperature Std dev of local std dev of cloud Gradient of cloud-top brightness temperature Gray-image texture (max angular second moment)

estimates derived from low-orbiter satellite PMW and IR data as a means to transport the PMW precipitation features during periods when microwave data are not available at a given location (Joyce et al. 2004). The Tropical Rainfall Measuring Mission (TRMM) Multi-satellite Precipitation Analysis (TMPA) blended IR information and PMW estimates, as well as available rain gauge analyses, to produce the final product with a calibration traceable back to the single “best” satellite estimate (Huffman et al. 2007). The Precipitation Estimation from Remotely Sensed Information Using Artificial Neural Networks (PERSIANN) product takes advantage of machine learning techniques to estimate precipitation rates with features extracted from IR grids and a window of grids surrounding them (Hsu et al. 1997). Similarly, the PERSIANN Cloud Classification System (PERSIANN-CCS), a revised PERSIANN product with finer resolution, also applies artificial neural networks to classify clouds based on IR information and then estimate precipitation (Hong et al. 2004). Other satellite-based precipitation products include the PMW-calibrated IR algorithm (PMIR; Kidd et al. 2003), the Self-Calibrating

Multivariate Precipitation Retrieval (SCaMPR) algorithm (Kuligowski 2002), and the Naval Research Laboratory Global Blended-Statistical Precipitation Analysis (NRLgeo; Turk and Miller 2005).

Despite the efforts of linking multisatellite information to surface precipitation, the accuracy of satellite-based products still remains insufficient (Boushaki et al. 2009). To deal with this problem, a variety of bias correction methods have been developed, mainly by incorporating additional available datasets, such as rain gauge or radar information (Boushaki et al. 2009; McCollum et al. 2002). However, ground measurements are only available in specific regions with a sufficient number of instruments. Therefore, several proposed bias correction methodologies are limited to a regional scale and are very difficult to extend to global applications. On the other hand, research also requires more satellite datasets to help reduce biases in the products. For instance, Behrangi et al. (2009) used multispectral data from the Geostationary Operational Environmental Satellite (GOES) and proved their effectiveness in precipitation detection. Li et al. (2007) and Nasrollahi

TABLE 2. Description of verification measures used.

Verification measures	Formulas	Descriptions	Remarks
POD	$POD = \frac{TP}{TP + MS}$	Fraction of events that are classified correctly	Where TP is the number of true positive events, MS is the number of missing events, FP is the number of false positive events, and TN is the number of true negative events
FAR	$FAR = \frac{FP}{TP + MS}$	Fraction of estimated events (positive) that are misclassified	
HSS	$HSS = \frac{2(TP \times TN - FP \times MS)}{(TP + MS)(MS + TN) + (TP + FP)(FP + TN)}$	Fractional improvement of the forecast over the standard forecast	
FBI	$FBI = \frac{TP + FP}{TP + MS}$	Bias of the binary forecast	
Bias	$bias = \bar{x} - \bar{y}$	Differences between estimations and observations over validation period	Where \bar{x} is the estimation average, \bar{y} is the observation average, x_i is the pixel estimation, y_i is the pixel observation, and N is the amount of observations
MSE	$MSE = \frac{1}{N} \sum (x_i - y_i)^2$	Closeness between estimations and observations over validation period	
Variance	$variance = MSE - bias^2$	Random errors within the estimation	

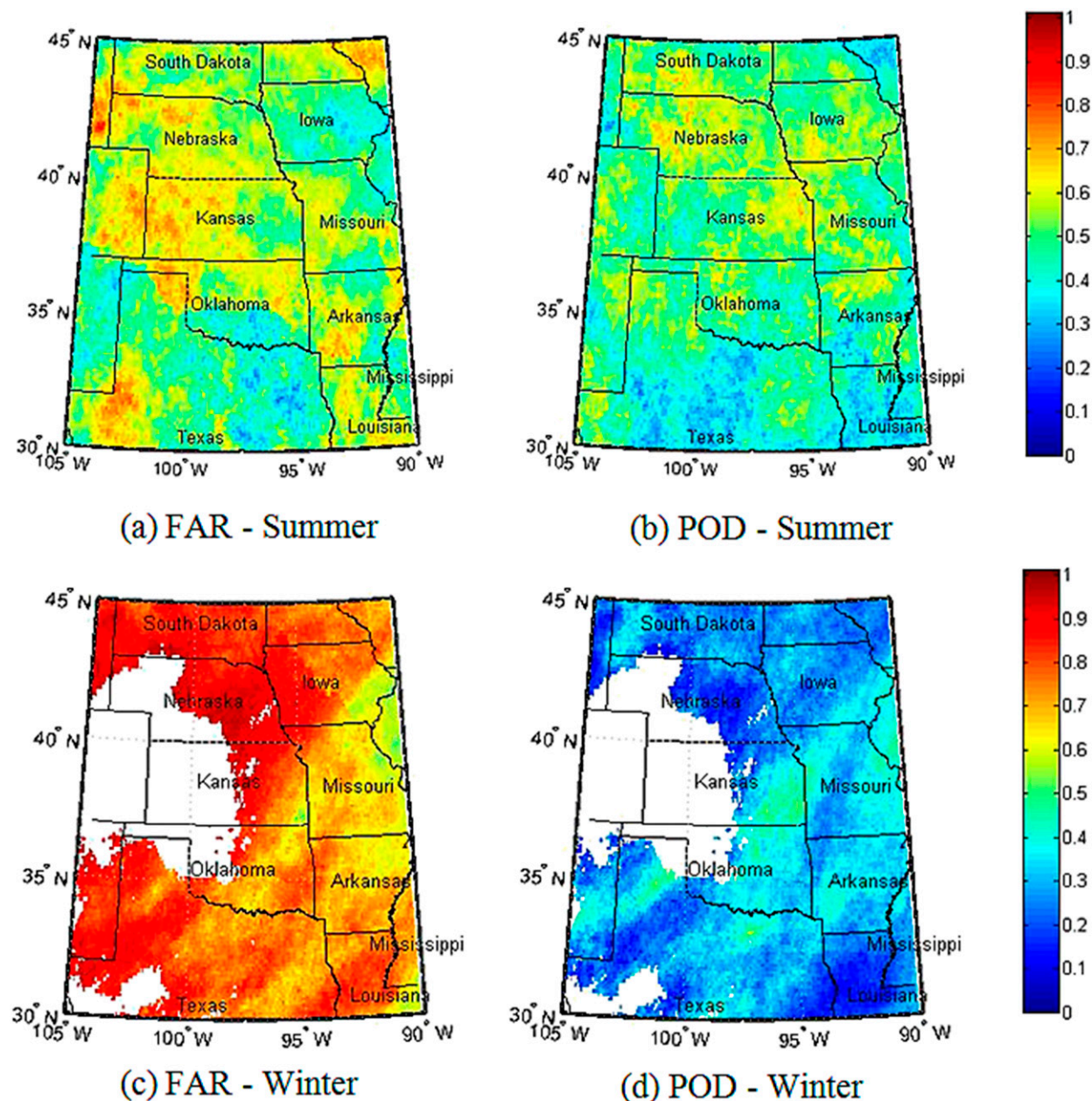


FIG. 1. FAR and POD of the PERSIANN-CCS precipitation data over the central United States (30°–45°N, 90°–105°W): (a),(b) summer (June–August 2013) and (c),(d) winter (from December 2013 to February 2014). The threshold used is 0.1 mm h⁻¹. The white color means that less than 50 precipitation pixels in the location are observed within corresponding periods.

et al. (2013) showed the value of the Moderate Resolution Imaging Spectroradiometer (MODIS) in identifying high clouds and thus reducing false alarms.

However, several studies emphasize that the key to making the best use of these datasets is promoting advanced methods that assist in the extraction of valuable information from the raw data (Nasrollahi et al. 2013; Sorooshian et al. 2011). In recent years, multiple novel techniques for deep learning have been developed in

the scientific discipline of machine learning, which is a breakthrough for dealing with large and complex datasets, especially for feature extraction from a large amount of image data (Bengio 2009; Hinton et al. 2006). The techniques have proved to be effective in dealing with many real-world data mining problems (Glorot et al. 2011a; Hinton et al. 2006; Lu et al. 2013; Vincent et al. 2008; Zhang et al. 2015). One particular advantage of a deep neural network (DNN) is that it helps extract

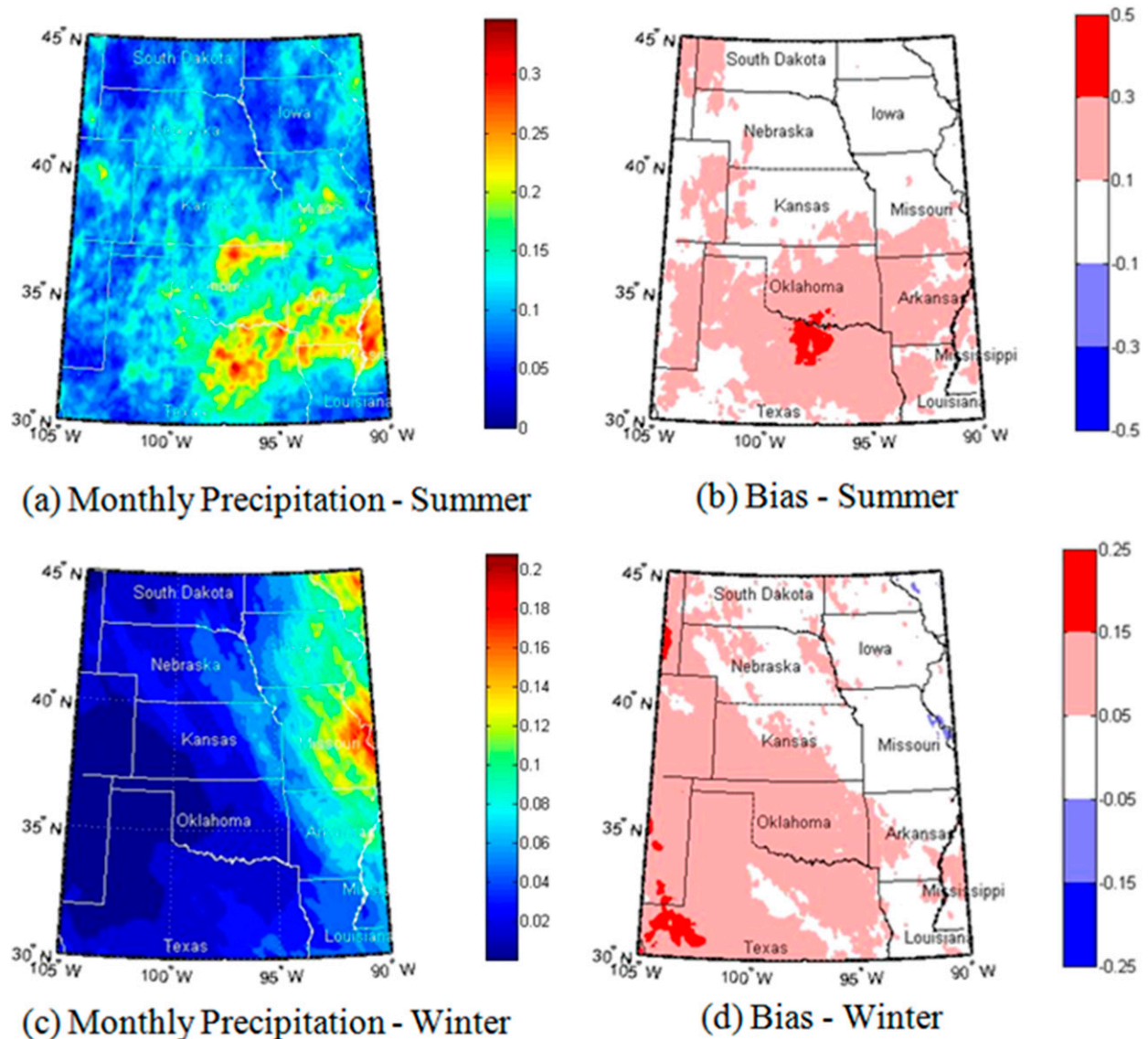


FIG. 2. Monthly precipitation observation (mm h^{-1}) and averaged bias (mm h^{-1}) of PERSIANN-CCS precipitation data over the central United States (30° – 45°N , 90° – 105°W): (a),(b) summer (June–August 2013) and (c),(d) winter (from December 2013 to February 2014). The ranges of the biases are from -0.5 to 0.5 mm h^{-1} for summer and from -0.25 to 0.25 mm h^{-1} for winter.

representative features automatically and further assists estimation. The power of deep learning for image processing and feature extraction provides an opportunity to improve the accuracy of satellite precipitation estimation.

As an initial step, we incorporate a modern DNN, stacked denoising autoencoder (SDAE), to perform the following tasks: 1) develop a bias correction system focusing on overestimation and false alarms, with a case study on the PERSIANN-CCS product; 2) demonstrate the effectiveness of deep learning for precipitation information extraction from satellite infrared imagery without adding any extra data from other sources; and

3) evaluate and analyze the case study results in the summer and winter seasons, respectively.

The remainder of this paper is organized as follows. Section 2 illustrates the bias in satellite precipitation estimation with a focus on false alarms and overestimations. Specific deep learning techniques are introduced and explained in detail in section 3. Section 4 describes the experimental design for the study, including the data used and model setup. Section 5 presents a comparison between the output of this study and the original satellite product. Finally, the main conclusions are summarized in section 6.

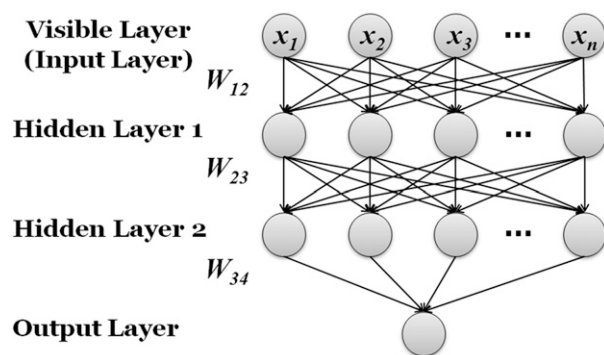


FIG. 3. A four-layer, fully connected DNN. The first layer is the visible/input layer, where the information is known. The second two layers are hidden layers, where each node is called a hidden node. The last layer is the output layer, which directly links to the target value that the model attempts to predict.

2. Bias in satellite precipitation estimation

One source of the bias in satellite precipitation estimation is that most precipitation products fail to extract the maximum amount of useful precipitation information buried in the satellite imagery. For example, a few statistics of an IR image, such as mean and standard deviation of nearby pixels, do not provide as much information as the raw image itself, where the full cloud-shape information is contained. In addition, some critical assumptions within the algorithms also lead to bias that cannot be ignored. For instance, PERSIANN-CCS has the assumption in the regression step that higher precipitation corresponds to lower brightness temperature T_b for pixels in a cloud patch. However, in reality, for convective storms, heavy precipitation

events actually occur at the edges of the cloud patch, where the T_b is not necessarily lower than that inside the patch. Various validation studies have been conducted to address the errors in satellite-based precipitation products and to investigate potential approaches to improve the algorithms (AghaKouchak et al. 2011; Bellerby and Sun 2005; Moazami et al. 2014; Tian et al. 2009). Overestimation with many false alarms is identified as a common drawback for most satellite-based precipitation products, especially in warm seasons (Sapiano and Arkin 2009). In addition, precipitation from low clouds is often missed in satellite-based products (Behrangi et al. 2009; Nasrollahi et al. 2013).

To investigate the bias in satellite-based products, common validation measurements are used to assess PERSIANN-CCS (Hong et al. 2004) with the National Centers for Environmental Prediction (NCEP) stage IV radar and gauge precipitation data (Baldwin and Mitchell 1996; Lin and Mitchell 2005). PERSIANN-CCS and NCEP stage IV data are compared at $0.08^\circ \times 0.08^\circ$ (latitude \times longitude) and hourly resolutions. The PERSIANN-CCS algorithm (Hong et al. 2004) manually designed and extracted nine features (Table 1) from IR cloud patches and applied an artificial neural network to estimate precipitation from these features. Table 2 gives the specific definitions of the validation measurements used in this paper.

Figure 1 shows the false alarm ratio (FAR) and probability of detection (POD) of both summer (June–August) of 2013 and winter (December–February) of 2013/14 over the central United States (30° – 45° N, 90° – 105° W), mainly including the states of Nebraska, Iowa, Kansas, Missouri, Oklahoma, Arkansas, and Texas.

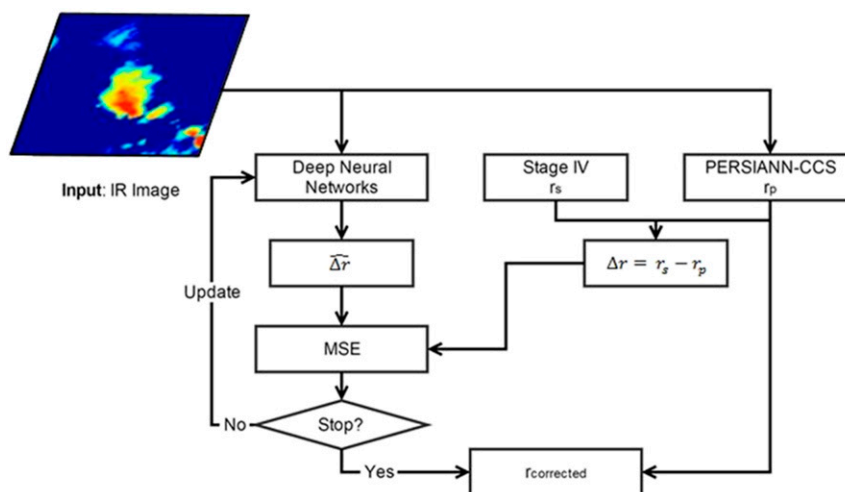


FIG. 4. Experimental design process. The input to the neural network is the IR image, and the output is the difference between the PERSIANN-CCS estimates and the NCEP stage IV measurements.

TABLE 3. Training, test, and validation period information and corresponding basic statistics for the warm season.

	Dataset	
	Training and test	Validation
Period	June–August 2012	June–August 2013
Sample	1 680 080	2 305 141
Avg precipitation (mm h^{-1})		
PERSIANN-CCS	0.100	0.185
NCEP stage IV	0.077	0.116
Max precipitation (mm h^{-1})		
PERSIANN-CCS	55.0	75.2
NCEP stage IV	85.1	112.0

High FARs can be observed for these regions, especially in the winter season, which is consistent with the results of previous studies (Nasrollahi et al. 2013; Tian et al. 2009). Similarly, POD is higher for summer than winter, while the number of precipitation events is fewer for winter than summer. Maps of monthly averaged precipitation rates and bias of the product are shown in Fig. 2. In the bias maps, the red colors indicate overestimation, while the blue colors indicate underestimation. For both seasons, the majority of the area shows overestimation, and few underestimation pixels could be detected. In addition, the warm season shows more overestimation compared to the cold season, though it has higher POD and lower FAR. Moreover, the region of the high bias corresponds to the region with relatively high monthly precipitation.

3. Methodology

This study develops a DNN framework that is capable of extracting “deeper” information automatically and effectively from satellite IR imagery to reduce bias in satellite precipitation products. One significant difference between DNNs and traditional artificial neural networks is that DNNs aim to automatically extract information at multiple levels of abstraction to allow a system to learn a complicated functional mapping of the

input to the output directly from the data while traditional neural networks tend to use manually designed features. It is achieved by applying the pretraining techniques to initialize weights to preserve information that better reconstructs the raw data (Bengio 2009). A more complete overview of the development of DNNs can be found in Bengio (2009).

Figure 3 presents a four-layer, fully connected artificial neural network, as used in this study. The network consists of neurons (or nodes) organized in layers through connections between nodes. A node receives inputs from connections, sums them, and passes the summation through a transformation function (or activation function) to produce an output delivered to nodes in the next layer. In the network, nodes in the first (top) layer receive input data; nodes in the last layer send outputs. The layers between input and output layers are called hidden layer(s). Connections between nodes have different strength or weights to determine various input–output relationships. To possess a required functional mapping, a deep architecture must assign a specific value to each weight (parameters); this is accomplished automatically by training the network with available input and output data samples.

Training techniques are essential in order for deep architectures to avoid getting stuck in local optima with poor performance (Bengio et al. 2007). SDAE, a widely used technique to train DNNs that was introduced by Vincent et al. (2008, 2010), has shown the capability to learn useful high-level representations from natural image patches and has thus been applied in several image recognition and other data mining studies (Glorot et al. 2011a; Lu et al. 2013; Xie et al. 2012; Zhou et al. 2012).

By taking advantage of this machine learning framework, valuable information is extracted from the input data, which helps to improve estimation. The method involves: 1) initializing parameters with an unsupervised greedy layer-wise pretraining process and 2) fine-tuning parameters of all layers globally to minimize a loss

TABLE 4. Training, test, and validation period information and corresponding basic statistics for the cold season.

	Dataset	
	Training and test	Validation
Period	From December 2012 to February 2013	From December 2013 to February 2014
Sample	820 776	902 037
Avg precipitation (mm h^{-1})		
PERSIANN-CCS	0.104	0.100
NCEP stage IV	0.080	0.040
Max precipitation (mm h^{-1})		
PERSIANN-CCS	51.2	62.56
NCEP stage IV	48.4	34.1

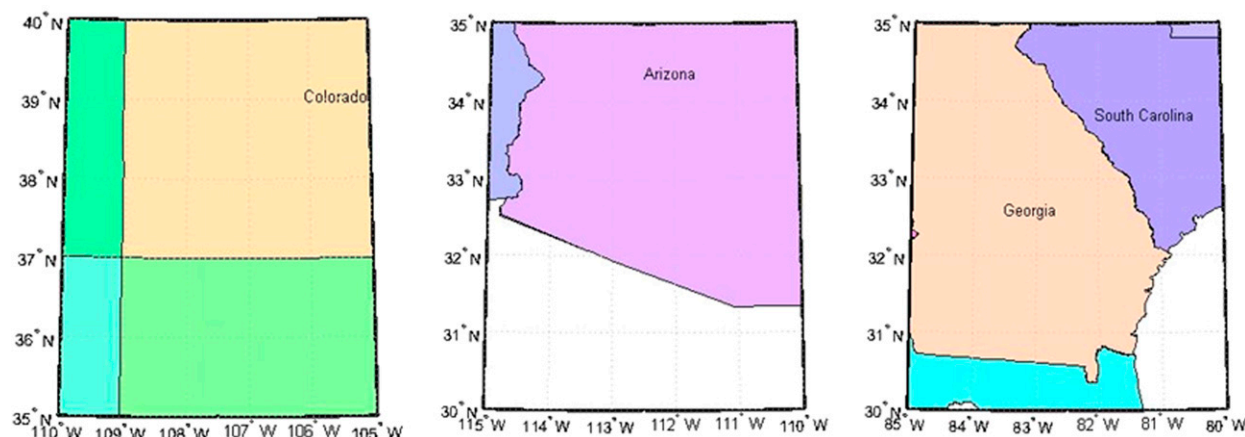


FIG. 5. Maps of selected additional validation regions.

function. A brief description of the SDAE is presented next. A more detailed explanation of the method can be found in [appendix A](#).

4. Experimental design

The design of the process is presented in [Fig. 4](#). In this study, the input data for the DNN are IR imagery collected by GOES, the same raw information used by PERSIANN-CCS. The dataset is at a spatial resolution of $0.08^\circ \times 0.08^\circ$ and an hourly temporal resolution. IR imagery provides cloud-top brightness temperature and has been used for multiple near-real-time precipitation estimation products ([Hong et al. 2004](#); [Hsu et al. 1997](#); [Huffman et al. 2007](#); [Joyce et al. 2004](#)). In PERSIANN-CCS, nine features of IR imagery in a cloud patch ([Table 1](#)) are used to predict precipitation rates at the (target) pixels within the cloud patch. Instead of using cloud image features designed by researchers, we allow the neural network to extract a useful representation for precipitation estimation itself. As shown in [Fig. 4](#), the input to DNN is a matrix $\mathbf{T}_{15 \times 15}$ containing the IR image in a 15×15 pixel window centered in pixel $t_{8,8}$, at which PERSIANN-CCS indicates a positive precipitation rate r_p . To produce a training data pool, the window is moved across the image sequentially, shifting location one grid box at a time in each hourly IR image of the study region.

The outputs/targets (at the same spatiotemporal resolutions as the input data) are the differences between

the PERSIANN-CCS estimates and the NCEP stage IV observations. The output is the value of the centered pixel of the 15×15 pixel window. That is, it is the adjusted quantity needed in order for the PERSIANN-CCS estimate to match the NCEP stage IV observed precipitation rate r_s at pixel $t_{8,8}$ ($\Delta r = r_p - r_s$). The reason we choose the differences, instead of directly estimating the NCEP stage IV precipitation rates, is that PERSIANN-CCS, as well as other satellite-based products, has successfully screened out a large number of no-rain (NR) pixels ([Hong et al. 2007](#)). Therefore, the input data are much more balanced and thus benefit the training process. However, the disadvantage of this design is that it will not help in reducing missing cases in PERSIANN-CCS. To cope with it, we are currently working on adjusting the model design to estimate precipitation directly instead of the difference. In addition, both inputs and outputs of the training data are normalized before training to shrink the range of the quantity and make it easier to operate.

After properly trained, DNN will produce $\widehat{\Delta r}$ (estimated difference) given IR imagery, without information from NCEP stage IV and PERSIANN-CCS. With this property, the model can potentially apply to areas without NCEP stage IV information and thus can offer global bias correction. On the other hand, the use of Δr (real difference) is during the training process, which optimizes the mean-square error (MSE) between $\widehat{\Delta r}$ and Δr as shown in [Fig. 4](#). Last, when producing

TABLE 5. Hyperparameters considered for SDAE in the study.

Hyperparameter	Description	Considered value
n_{HLayers}	Number of hidden layers	1, 2, and 3
n_{HNodes}	Number of hidden nodes	500, 1000, and 2000
$p_{\text{Corruption}}$	Percentage of corruption for pretraining	0.1, 0.2, and 0.4

TABLE 6. The R/NR classification performance of PERSIANN-CCS and DNN-corrected precipitation. Values in parentheses are the relative performance of DNN corrected and PERSIANN-CCS.

	Summer	Winter
No. of precipitation pixels (h^{-1})		
NCEP stage IV	533	358
PERSIANN-CCS	694 (30% more)	681 (90% more)
DNN corrected	516 (3% less)	409 (14% more)
No. of false positive pixels (h^{-1})		
PERSIANN-CCS	395	598
DNN corrected	264 (33% corrected)	339 (43% corrected)
No. of misclassified pixels ^a (h^{-1})		
PERSIANN-CCS	629	873
DNN corrected	545 (13% corrected)	627 (28% corrected)
HSS		
PERSIANN-CCS	0.478	0.148
DNN corrected	0.473	0.174
FBI		
PERSIANN-CCS	1.27	1.86
DNN corrected	0.93	1.09

^a Includes both false positive (false alarm) and false negative (missing) events.

adjusted precipitation, all negative values will be forced to zero.

The study periods are the summer (June–August) of 2012 and 2013 and winter (December–February) of 2012/13 and 2013/14. The study area is the central United States (30° – 45°N , 90° – 105°W), as described in section 2. The summer and winter seasons are treated separately to account for different climate conditions and to improve model accuracy. To properly validate the methodology, we divided the data into training, test,

and validation datasets for the summer and winter seasons, respectively. During the training process, the training and test datasets are used to calibrate the parameters and prevent overfitting. They are randomly selected from the same time periods (summer 2012 and winter 2012/13, respectively) in a ratio of 75:25. On the other hand, validation datasets for SDAE performance evaluation are taken from the same seasons of the next year. More detailed information and some basic statistics of the training, test, and validation datasets of both seasons are shown in Tables 3 and 4. The number of sample points in training and test datasets for both seasons are considerable and are expected to accommodate the automatic feature-extraction process. Moreover, to demonstrate the effectiveness of the methodology with variability in space, three regions with NCEP stage IV measurements outside the study area are selected for additional validation. These regions are 1) the Colorado area (35° – 40°N , 105° – 110°W), 2) the Arizona area (30° – 35°N , 110° – 115°W), and 3) the Georgia area (30° – 35°N , 80° – 85°W). The maps of the regions are shown in Fig. 5.

The objective function of SDAE to set the optimal weight values is the MSE on the output. In addition, in this study, we use the rectified linear activation function, which is a most popular choice for real-value estimation (Glorot et al. 2011b). After various combinations were tested and compared, a four-layer neural network with 1000 hidden nodes for each hidden layer with 40% input corruption in the training was selected for this study. Table 5 gives some common hyperparameters needed for training a DNN with SDAE. The choices considered

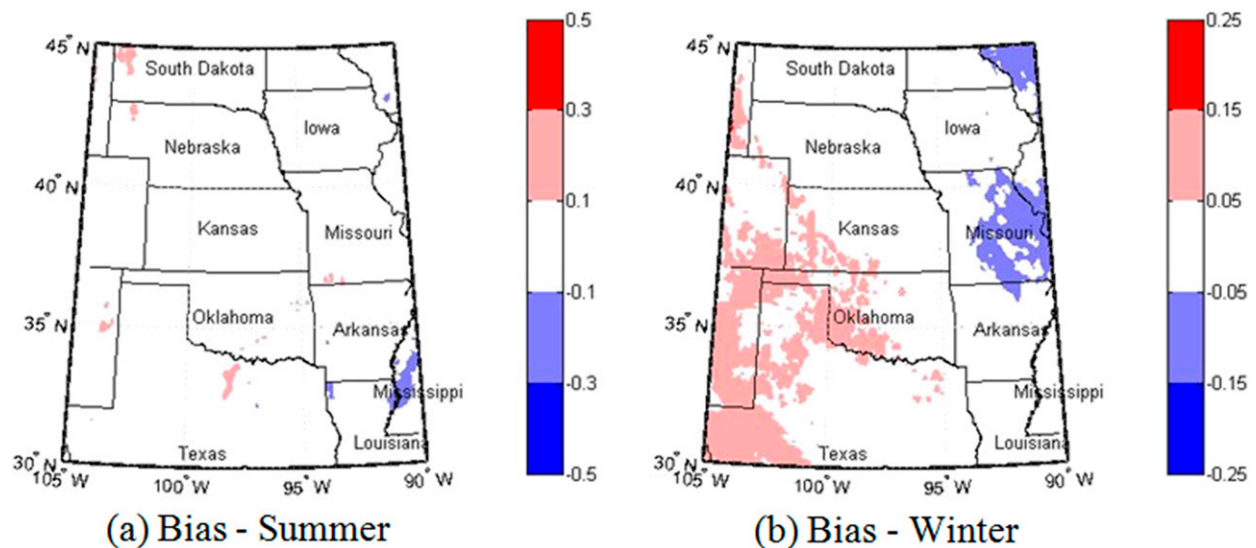


FIG. 6. Averaged bias (mm h^{-1}) of DNN-corrected output over the central United States (30° – 45°N , 90° – 105°W): (a) summer (June–August 2013) and (b) winter (from December 2013 to February 2014). The ranges of the biases are from -0.5 to 0.5 mm h^{-1} for summer and from -0.25 to 0.25 mm h^{-1} for winter.

TABLE 7. Averaged bias, variance, and MSE of PERSIANN-CCS and DNN-corrected precipitation. Values in parentheses are the relative performance of DNN corrected and PERSIANN-CCS.

	Summer	Winter
Averaged bias (mm h^{-1})		
PERSIANN-CCS	0.091	0.054
DNN corrected	0.002 (98% corrected)	0.012 (78% corrected)
Averaged variance [$(\text{mm h}^{-1})^2$]		
PERSIANN-CCS	2.330	0.442
DNN corrected	1.596 (31% corrected)	0.306 (31% corrected)
Averaged MSE [$(\text{mm h}^{-1})^2$]		
PERSIANN-CCS	2.338	0.445
DNN corrected	1.596 (32% corrected)	0.306 (31% corrected)

in this study are typical choices for the corresponding parameters (Vincent et al. 2010). Note that, here, we only considered neural networks with an equal number of hidden nodes at all hidden layers for processing convenience. The result should not be fundamentally different from other possible combinations. Other potential hyperparameters, such as learning rate and training iterations, were decided manually to optimize the result within the training process.

5. Results and discussion

The results presented here show the performances of the SDAE model in the validation periods (summer of 2013 and winter of 2013/14) in comparison with the original PERSIANN-CCS data. The evaluation includes both detection of rain/no-rain (R/NR) pixels and intensity of the precipitation amount for warm and cold seasons, respectively. In addition, as an example, results of the rainfall event on 4 August 2014 are analyzed and compared with both PERSIANN-CCS estimation and NCEP stage IV observation. In this section, “DNN corrected” refers to the bias-corrected precipitation using SDAE.

Table 6 provides the binary R/NR detection performance of PERSIANN-CCS and DNN-corrected precipitation, including the averaged hourly number of precipitation pixels, false positive pixels, and misclassified pixels. The performance is evaluated for hourly estimation on the study area and averaged over the validation periods for warm and cold seasons separately. The bias correction process is very effective at identifying false alarm pixels and balancing the number of precipitation pixels. Specifically, the averaged hourly number of false alarm pixels drops from 395 to 264 and from 598 to 339 for summer and winter (i.e., 33% and 43% correction), respectively. The model properly reduced the overestimation of the number of pixels with precipitation in PERSIANN-CCS (30% more to 3% less and 90% more to 14% more relative to NCEP stage IV

observations). The overall number of misclassified pixels is reduced for both warm and cold seasons (i.e., 13% and 28% correction), respectively. The Heidke skill score (HSS) of DNN corrected is similar to PERSIANN-CCS in summer and slightly better in winter. The model’s incapability of dealing with missing cases may prevent it from improving the score and thus shows the necessity of moving on to a DNN that directly estimates precipitation from IR imagery. Moreover, the frequency bias (FBI) shows that the forecast biases are reduced for both seasons compared to PERSIANN-CCS. This suggests that the model is capable of identifying false alarm pixels in the original PERSIANN-CCS. Raw R/NR results on the validation periods can be found in appendix B.

Figure 6 presents maps of the bias of DNN-corrected precipitation over the study region averaged in the warm and cold validation periods, respectively, which is the same region and time used in Fig. 2. The white color indicates very small bias and shows that the DNN model has made relatively significant corrections to the PERSIANN-CCS precipitation pixels, especially in the summer season. The overestimation produced by the PERSIANN-CCS product is mostly removed. Specific calculations are displayed in Table 7. The averaged biases are only 0.002 and 0.012 mm day^{-1} after bias correction, compared to 0.091 and 0.054 mm day^{-1} before bias correction for summer and winter (i.e., 98% and 78% correction), respectively.

Similar results can be seen in Fig. 7, which shows the MSE of PERSIANN-CCS and DNN-corrected precipitation over the study region, averaged over the validation periods. The warm colors indicate strong differences compared to NCEP stage IV observations, while the cold colors indicate small differences. The heavy errors shown in PERSIANN-CCS over the summer validation period (Fig. 7a) are strongly reduced by the model (Fig. 7b). Similar results can be observed for the winter period (Figs. 7c,d). However, as Table 7 illustrates, over 30% correction in averaged MSE is observed for both seasons,

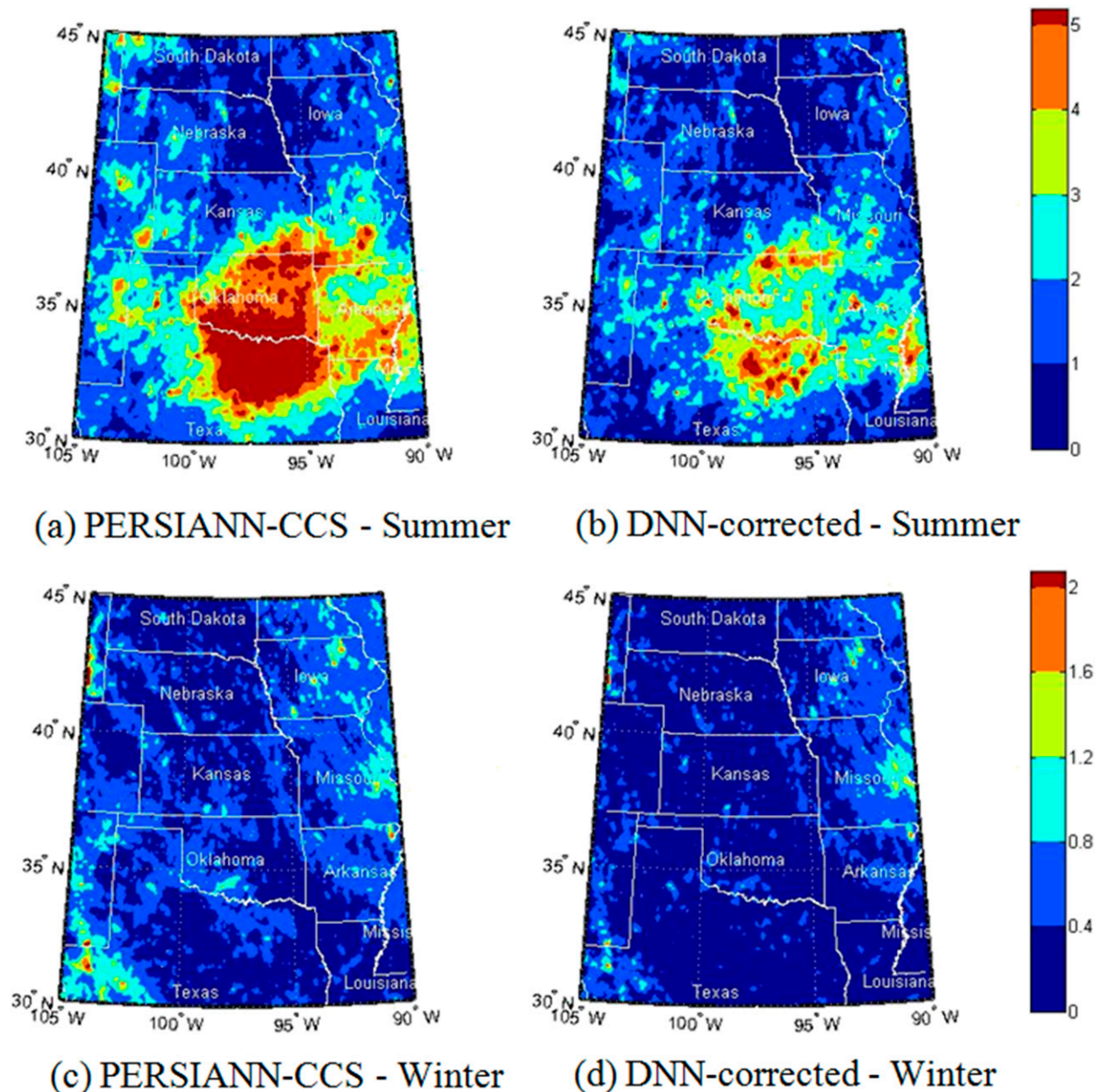


FIG. 7. Averaged MSE $[(\text{mm h}^{-1})^2]$ of (left) PERSIANN-CCS and (right) DNN-corrected output over the central United States (30° – 45° N, 90° – 105° W): (a),(b) summer (June–August 2013) and (c),(d) winter (from December 2013 to February 2014).

and the absolute improvement in summer is more significant. The results indicate the model's ability to correct the bias of overall precipitation intensity for both warm and cold seasons by automatically extracting useful features from satellite data.

To demonstrate how SDAE has significantly improved estimates of individual precipitation events, Fig. 8 and Table 8 present the analysis of a rainfall event on 4 August 2014. The event is randomly selected from noticeable rainfall events within the validation periods.

The cumulative amounts of the rainfall event for PERSIANN-CCS, DNN-corrected, and NCEP stage IV precipitation are all displayed in Fig. 8. It can be seen that overestimation in the original PERSIANN-CCS is reduced remarkably, while the rainfall distribution pattern is also adjusted toward the observation to some extent. This effect is quantified in Table 8 for both R/NR detection and intensity. As for detection performance, the number of precipitation pixels in PERSIANN-CCS is reduced from 22% overestimated to just 2%, while

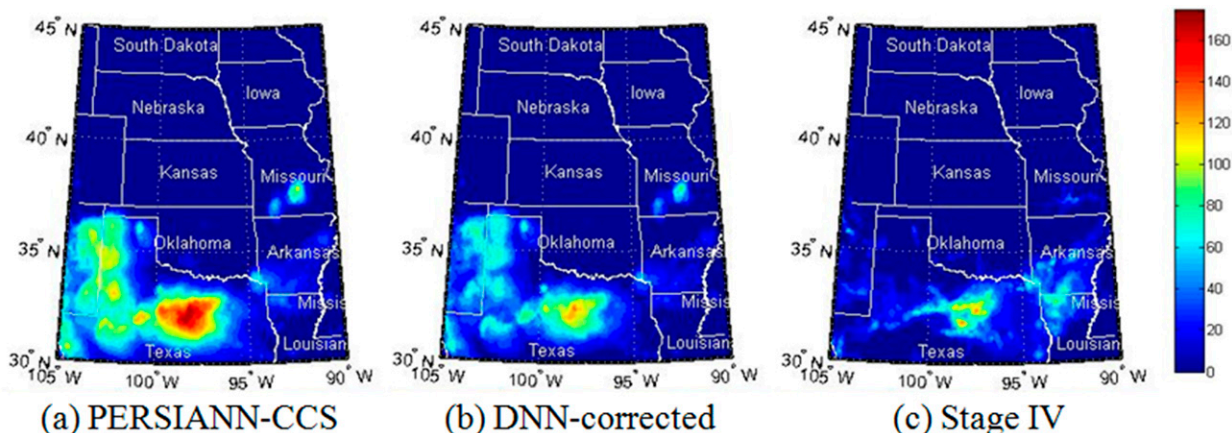


FIG. 8. Cumulative precipitation amounts (mm day^{-1}) of PERSIANN-CCS estimation, DNN-corrected estimation, and NCEP stage IV observation on 4 Aug 2014, over the central United States (30° – 45°N , 90° – 105°W): (a) PERSIANN-CCS, (b) DNN corrected, and (c) NCEP stage IV.

around 23% of false positive pixels are corrected. Raw R/NR results on the validation periods can be found in appendix B. As for intensity, averaged bias and MSE decrease from 0.398 and 8.267 to 0.164 and 4.875 (i.e., 58% and 41% correction), respectively. This example demonstrates the effectiveness of the model to help improve precipitation estimation for typical storm events. Meanwhile, notice that the scheme is unable to deal with the missing precipitation of the original PERSIANN-CCS. An area of future research for us will be to apply the method to direct precipitation estimation to start addressing this issue.

To validate the effectiveness of the methodology when the coefficients are applied in other locations, Table 9 summarizes averaged bias, variance, and MSE of DNN-corrected and PERSIANN-CCS precipitation over three areas outside of the study region on the warm and cold validation periods, respectively. Generally, the model works effectively to reduce bias and variance of the original PERSIANN-CCS. For the Colorado area (35° – 40°N , 105° – 110°W) and the Arizona area (30° – 35°N , 110° – 115°W), MSEs are improved at least 29% for both warm and cold seasons, while improvement is only around 4% for summer for the Georgia area (30° – 35°N , 80° – 85°W). One possible reason for this is that the original PERSIANN-CCS has a relatively large amount of missing and underestimation in this area. Therefore, our model will not be helpful in those situations, as discussed above.

6. Conclusions

The aim of this paper is to apply a deep neural network framework to satellite-based precipitation estimation products to correct the estimation bias in a data-driven manner by extracting more useful features from satellite

imagery. More specifically, SDAE, a popular technique in image recognition, is employed to improve the PERSIANN-CCS product. The model is trained in 2012–13 and evaluated during the 2013 summer and 2013/14 winter seasons.

Verification studies show improved results in both R/NR detection and precipitation intensity over the validation period for both seasons. Binary R/NR detection resulted in the correction of a significant number of false alarm pixels, especially in the cold season. For precipitation intensity, the averaged daily biases are corrected by as much as 98% and 78% in the validation warm and cold seasons, respectively. These results are also illustrated for a specific rainfall event on 4 August 2014, for which visualization of the cumulative rainfall

TABLE 8. Performance of PERSIANN-CCS and DNN-corrected precipitation on 4 Aug 2014. Values in parentheses are the relative performance of DNN corrected and PERSIANN-CCS.

No. of precipitation pixels (h^{-1})	
NCEP stage IV	1433
PERSIANN-CCS	1732 (22% more)
DNN corrected	1450 (2% more)
No. of false positive pixels (h^{-1})	
PERSIANN-CCS	805
DNN corrected	622 (23% corrected)
No. of misclassified pixels ^a (h^{-1})	
PERSIANN-CCS	1311
DNN corrected	1227 (6% corrected)
Averaged bias (mm h^{-1})	
PERSIANN-CCS	0.398
DNN corrected	0.164 (58% corrected)
Averaged MSE [$(\text{mm h}^{-1})^2$]	
PERSIANN-CCS	8.267
DNN corrected	4.875 (41% corrected)

^a Includes both false positive (false alarm) and false negative (missing) events.

TABLE 9. Averaged bias, variance, and MSE of PERSIANN-CCS and DNN-corrected precipitation over areas outside of the study region. Values in parentheses are the relative performance of DNN corrected and PERSIANN-CCS.

		Summer	Winter
Colorado area			
Averaged bias (mm h^{-1})	PERSIANN-CCS	0.091	0.062
	DNN corrected	0.002 (99% corrected)	−0.005 (92% corrected)
Averaged variance [$(\text{mm h}^{-1})^2$]	PERSIANN-CCS	0.738	0.722
	DNN corrected	0.498 (32% corrected)	0.421 (42% corrected)
Averaged MSE [$(\text{mm h}^{-1})^2$]	PERSIANN-CCS	0.746	0.725
	DNN corrected	0.498 (33% corrected)	0.421 (42% corrected)
Arizona area			
Averaged bias (mm h^{-1})	PERSIANN-CCS	0.028	0.085
	DNN corrected	0.004 (86% corrected)	0.002 (98% corrected)
Averaged variance [$(\text{mm h}^{-1})^2$]	PERSIANN-CCS	0.331	0.778
	DNN corrected	0.235 (29% corrected)	0.437 (44% corrected)
Averaged MSE [$(\text{mm h}^{-1})^2$]	PERSIANN-CCS	0.332	0.785
	DNN corrected	0.235 (29% corrected)	0.437 (44% corrected)
Georgia area			
Averaged bias (mm h^{-1})	PERSIANN-CCS	−0.039	0.017
	DNN corrected	−0.052 (33% worse)	−0.016 (6% corrected)
Averaged variance [$(\text{mm h}^{-1})^2$]	PERSIANN-CCS	0.911	0.862
	DNN corrected	0.875 (4% corrected)	0.655 (24% corrected)
Averaged MSE [$(\text{mm h}^{-1})^2$]	PERSIANN-CCS	0.911	0.862
	DNN corrected	0.878 (4% corrected)	0.657 (24% corrected)

amount demonstrates the model's ability to correct false alarms and overestimation.

The results verify that useful information is available in IR imagery and can help improve the quality of satellite precipitation products with respect to detecting R/NR pixels and quantifying the precipitation rates. More important, such useful information for precipitation estimation can be extracted automatically by deep neural networks. Moreover, the methodology can be easily integrated into near-real-time operational precipitation estimation products and can help extract additional features from satellite datasets to reduce bias. Meanwhile, the application of the technique is not limited to IR imagery, but should be extendable to multiple satellite datasets because of its ability to automatically extract information. The case study of PERSIANN-CCS proves its advantage compared to a few manually designed features.

In addition, our results suggest that GOES cloud IR imagery still contains valuable information that has not been utilized by most satellite precipitation retrieval algorithms. Our experiment demonstrates that the cloud IR image from a 15×15 pixel window is more informative than the nine IR statistic features used in PERSIANN-CCS as the input data for precipitation estimation. Such information can be extracted automatically by a well-designed deep neural network. The next step for this work will be to explore the possibility of using deep learning techniques to produce a precipitation estimation product directly instead of bias correction.

Moreover, we believe that these data-driven methodologies can benefit many fields of weather forecasting, climate variability, hydrology, and water resources management.

Acknowledgments. Financial support for this study is made available from the NSF Cyber-Enabled Sustainability Science and Engineering (Grant CCF-1331915), the U.S. Army Research Office (Grant W911NF-11-1-0422), the NASA Precipitation Measurement Mission (Grant NNX13AN60G), and the NASA Earth and Space Science Fellowship (Grant NNX15AN86H).

APPENDIX A

SDAE Calibration Scheme

The SDAE involves 1) initializing parameters (i.e., weights) with an unsupervised greedy layer-wise pretraining process and 2) fine-tuning parameters of all layers globally to minimize a loss function.

a. Greedy layer-wise pretraining

The denoising autoencoder (DAE) method is used as a layer-wise pretraining technique to initialize the weights for each layer of the DNN. Initialization is often a crucial component of successful algorithms for deep architectures (Bengio 2009). An autoencoder (AE), structured as in Fig. A1, is an unsupervised learning process that trains the weights to reconstruct the input \mathbf{x} using its internal representation (hidden nodes) \mathbf{h}

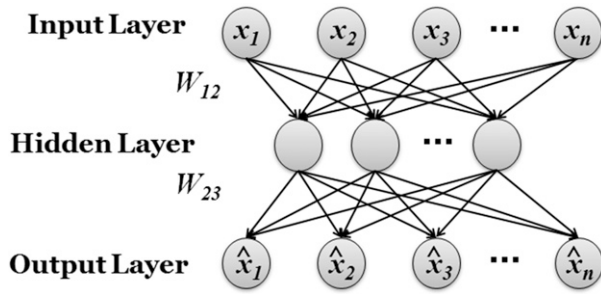


FIG. A1. An autoencoder. The structure is similar to a regular neural network as shown in Fig. 3, but its output layer is a reconstruction of the input layer. Therefore, AE is an unsupervised learning structure without a target value associated with it.

(Bourlard and Kamp 1988; Hinton and Zemel 1993). The representation is called the encoder:

$$\mathbf{h} = f(\mathbf{x}) = s(\mathbf{W}_{12}\mathbf{x} + \mathbf{b}_1),$$

where $f(\cdot)$ is a deterministic mapping function; s is a nonlinear operator, known as an activation function; \mathbf{W}_{12} denotes the weights between the input and hidden layers; and \mathbf{b}_1 is an offset vector. Common choices for s include the sigmoid function, hyperbolic tangent function, and rectified linear function. The loss function to minimize is typically a reconstruction error (RE) given the hidden nodes \mathbf{h} , such as the MSE:

$$\text{RE} = \|\mathbf{x} - \hat{\mathbf{x}}\|^2,$$

where $\hat{\mathbf{x}}$ is the reconstructed estimate of \mathbf{x} , given by

$$\hat{\mathbf{x}} = g(\mathbf{h}) = s(\mathbf{W}_{23}\mathbf{h} + \mathbf{b}_2),$$

where $g(\cdot)$, with a similar form to $f(\cdot)$, is called the decoder; \mathbf{W}_{23} denotes the weights between the hidden and output (reconstruction) layers; and \mathbf{b}_2 is an offset vector. In deep architectures, overcomplete (meaning that it is of a higher dimensional than on \mathbf{x}) but sparse representations are widely used to help extract useful sparse

features (Le et al. 2011; Lee et al. 2007; Ranzato et al. 2007).

However, AE methods with overcomplete representations are unable to guarantee the extraction of useful features because they can lead to the obvious solution: “simply copy the input” (Vincent et al. 2010). To overcome this problem, DAE extracts robust representations by reconstructing the input from a noisy version of it. As indicated by Vincent et al. (2008), the useful higher-level information should be rather stable and robust under perturbation of the input. One typical way to do so is called “masking noise,” which randomly forces a fraction of the input elements to be zero. For example, for an image, the input used will be a “broken” version of the raw image. Starting from the input layer, DAE is applied to initialize weights between layers sequentially, except the last layer, which is the output layer.

b. Supervised fine tuning

Learning the weights one layer at a time is computationally efficient but does not provide the optimal weights for the overall prediction task (Hinton et al. 2006). Instead, these weights can be treated as an initialization, followed by a traditional supervised learning process used to fine-tune all of the weights simultaneously to further optimize the whole neural network. The backpropagation (backward propagation of errors) algorithm, which performs gradient descent, is commonly used for training artificial neural networks to minimize the loss function (Rumelhart et al. 1986). MSE is used as the loss function in this project because it is a common choice for real-valued output. In this step, the complete raw input is used to produce higher levels of representations (Vincent et al. 2010).

APPENDIX B

Table B1 provides raw R/NR results on the validation periods and 4 August 2014.

TABLE B1. Raw R/NR classification results of PERSIANN-CCS and DNN-corrected precipitation of the validation periods and on 4 Aug 2014.

	True positive	False positive	False negative	True negative
Summer 2014 (h^{-1})				
PERSIANN-CCS	299	395	234	34 228
DNN corrected	252	264	281	34 359
Winter 2014 (h^{-1})				
PERSIANN-CCS	83	598	275	34 200
DNN corrected	70	339	288	34 459
4 Aug 2014 (h^{-1})				
PERSIANN-CCS	927	805	506	32 918
DNN corrected	828	622	605	33 101

REFERENCES

- AghaKouchak, A., N. Nasrollahi, J. Li, B. Imam, and S. Sorooshian, 2011: Geometrical characterization of precipitation patterns. *J. Hydrometeorol.*, **12**, 274–285, doi:10.1175/2010JHM1298.1.
- Baldwin, M. E., and K. E. Mitchell, 1996: The NCEP hourly multi-sensor U.S. precipitation analysis. Preprints, *11th Conf. on Numerical Weather Prediction*, Norfolk, VA, Amer. Meteor. Soc., J95–J96.
- Behrangi, A., K. L. Hsu, B. Imam, S. Sorooshian, G. J. Huffman, and R. J. Kuligowski, 2009: PERSIANN-MSA: A Precipitation Estimation Method from Satellite-Based Multispectral Analysis. *J. Hydrometeorol.*, **10**, 1414–1429, doi:10.1175/2009JHM1139.1.
- Bellerby, T. J., and J. Z. Sun, 2005: Probabilistic and ensemble representations of the uncertainty in an IR/microwave satellite precipitation product. *J. Hydrometeorol.*, **6**, 1032–1044, doi:10.1175/JHM454.1.
- Bengio, Y., 2009: Learning deep architectures for AI. *Found. Trends Mach. Learn.*, **2**, 1–127, doi:10.1561/22000000006.
- , P. Lamblin, D. Popovici, and H. Larochelle, 2007: Greedy layer-wise training of deep networks. *Advances in Neural Information Processing Systems 19*, B. Scholkopf, J. Platt, and T. Hoffman, Eds., MIT Press, 153–160.
- Bourlard, H., and Y. Kamp, 1988: Auto-association by multilayer perceptrons and singular value decomposition. *Biol. Cybern.*, **59**, 291–294, doi:10.1007/BF00332918.
- Boushaki, F. I., K. L. Hsu, S. Sorooshian, G. H. Park, S. Mahani, and W. Shi, 2009: Bias adjustment of satellite precipitation estimation using ground-based measurement: A case study evaluation over the southwestern United States. *J. Hydrometeorol.*, **10**, 1231–1242, doi:10.1175/2009JHM1099.1.
- Glorot, X., A. Bordes, and Y. Bengio, 2011a: Deep sparse rectifier neural networks. *J. Mach. Learn. Res.*, **15**, 315–323. [Available online at <http://jmlr.csail.mit.edu/proceedings/papers/v15/glorot11a/glorot11a.pdf>.]
- , —, and —, 2011b: Domain adaptation for large-scale sentiment classification: A deep learning approach. *Proceedings of the 28th International Conference on Machine Learning*, L. Getoor and T. Scheffer, Eds., Omnipress, 513–520. [Available online at http://www.icml-2011.org/papers/342_icmlpaper.pdf.]
- Hinton, G. E., and R. S. Zemel, 1993: Autoencoders, minimum description length and Helmholtz free energy. *Advances in Neural Information Processing Systems 6*, J. D. Cowan, G. Tesauero, and J. Alspector, Eds., Morgan Kaufmann, 3–10. [Available online at <http://papers.nips.cc/paper/798-autoencoders-minimum-description-length-and-helmholtz-free-energy>.]
- , S. Osindero, and Y. W. Teh, 2006: A fast learning algorithm for deep belief nets. *Neural Comput.*, **18**, 1527–1554, doi:10.1162/neco.2006.18.7.1527.
- Hong, Y., K. L. Hsu, S. Sorooshian, and X. G. Gao, 2004: Precipitation Estimation from Remotely Sensed Imagery Using an Artificial Neural Network Cloud Classification System. *J. Appl. Meteorol.*, **43**, 1834–1852, doi:10.1175/JAM2173.1.
- , D. Gochis, J. T. Cheng, K. L. Hsu, and S. Sorooshian, 2007: Evaluation of PERSIANN-CCS rainfall measurement using the NAME Event Rain Gauge Network. *J. Hydrometeorol.*, **8**, 469–482, doi:10.1175/JHM574.1.
- Hsu, K. L., X. G. Gao, S. Sorooshian, and H. V. Gupta, 1997: Precipitation Estimation from Remotely Sensed Information Using Artificial Neural Networks. *J. Appl. Meteorol.*, **36**, 1176–1190, doi:10.1175/1520-0450(1997)036<1176:PEFRSI>2.0.CO;2.
- Huffman, G. J., and Coauthors, 2007: The TRMM Multisatellite Precipitation Analysis (TMPA): Quasi-global, multiyear, combined-sensor precipitation estimates at fine scales. *J. Hydrometeorol.*, **8**, 38–55, doi:10.1175/JHM560.1.
- Joyce, R. J., J. E. Janowiak, P. A. Arkin, and P. P. Xie, 2004: CMORPH: A method that produces global precipitation estimates from passive microwave and infrared data at high spatial and temporal resolution. *J. Hydrometeorol.*, **5**, 487–503, doi:10.1175/1525-7541(2004)005<0487:CAMTPG>2.0.CO;2.
- Kidd, C., D. R. Kniveton, M. C. Todd, and T. J. Bellerby, 2003: Satellite rainfall estimation using combined passive microwave and infrared algorithms. *J. Hydrometeorol.*, **4**, 1088–1104, doi:10.1175/1525-7541(2003)004<1088:SREUCP>2.0.CO;2.
- Kuligowski, R. J., 2002: A self-calibrating real-time GOES rainfall algorithm for short-term rainfall estimates. *J. Hydrometeorol.*, **3**, 112–130, doi:10.1175/1525-7541(2002)003<0112:ASCRGT>2.0.CO;2.
- Le, Q. V., J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, and A. Y. Ng, 2011: On optimization methods for deep learning. *Proceedings of the 28th International Conference on Machine Learning*, L. Getoor and T. Scheffer, Eds., Omnipress, 265–272. [Available online at http://www.icml-2011.org/papers/210_icmlpaper.pdf.]
- Lee, H., C. Ekanadham, and A. Ng, 2007: Sparse deep belief net model for visual area V2. *Advances in Neural Information Processing Systems 20*, J. C. Platt et al., Eds., MIT Press, 8 pp. [Available online at <http://papers.nips.cc/paper/3313-sparse-deep-belief-net-model-for-visual-area-v2.pdf>.]
- Li, Z., J. Li, W. Menzel, T. Schmit, and S. Ackerman, 2007: Comparison between current and future environmental satellite imagers on cloud classification using MODIS. *Remote Sens. Environ.*, **108**, 311–326, doi:10.1016/j.rse.2006.11.023.
- Lin, Y., and K. E. Mitchell, 2005: The NCEP stage II/IV hourly precipitation analyses: Development and applications. *19th Conf. on Hydrology*, San Diego, CA, Amer. Meteor. Soc., 1.2. [Available online at http://ams.confex.com/ams/Annual2005/techprogram/paper_83847.htm.]
- Lu, X., Y. Tsao, S. Matsuda, and C. Hori, 2013: Speech enhancement based on deep denoising autoencoder. *INTERSPEECH 2013*, F. Bimbot et al., Eds., International Speech Communication Association, 436–440. [Available online at http://www.isca-speech.org/archive/interspeech_2013/13_0436.html.]
- Marzano, F. S., M. Palmacci, D. Cimini, G. Giuliani, and F. J. Turk, 2004: Multivariate statistical integration of satellite infrared and microwave radiometric measurements for rainfall retrieval at the geostationary scale. *IEEE Trans. Geosci. Remote Sens.*, **42**, 1018–1032, doi:10.1109/TGRS.2003.820312.
- McCollum, J. R., W. F. Krajewski, R. R. Ferraro, and M. B. Ba, 2002: Evaluation of biases of satellite rainfall estimation algorithms over the continental United States. *J. Appl. Meteorol.*, **41**, 1065–1080, doi:10.1175/1520-0450(2002)041<1065:EOBOSR>2.0.CO;2.
- Moazami, S., S. Golian, M. R. Kavianpour, and Y. Hong, 2014: Uncertainty analysis of bias from satellite rainfall estimates using copula method. *Atmos. Res.*, **137**, 145–166, doi:10.1016/j.atmosres.2013.08.016.
- Nasrollahi, N., K. L. Hsu, and S. Sorooshian, 2013: An Artificial Neural Network model to reduce false alarms in satellite precipitation products using MODIS and CloudSat observations. *J. Hydrometeorol.*, **14**, 1872–1883, doi:10.1175/JHM-D-12-0172.1.
- Ranzato, M. A., Y.-L. Boureau, and Y. LeCun, 2007: Sparse feature learning for deep belief networks. *Advances in Neural Information Processing Systems 20*, J. C. Platt et al., Eds., MIT

- Press, 8 pp. [Available online at <http://papers.nips.cc/paper/3363-sparse-feature-learning-for-deep-belief-networks.pdf>.]
- Rumelhart, D. E., G. E. Hinton, and R. J. Williams, 1986: Learning representations by back-propagating errors. *Nature*, **323**, 533–536, doi:[10.1038/323533a0](https://doi.org/10.1038/323533a0).
- Sapiano, M. R. P., and P. A. Arkin, 2009: An intercomparison and validation of high-resolution satellite precipitation estimates with 3-hourly gauge data. *J. Hydrometeor.*, **10**, 149–166, doi:[10.1175/2008JHM1052.1](https://doi.org/10.1175/2008JHM1052.1).
- Sorooshian, S., and Coauthors, 2011: Advanced concepts on remote sensing of precipitation at multiple scales. *Bull. Amer. Meteor. Soc.*, **92**, 1353–1357, doi:[10.1175/2011BAMS3158.1](https://doi.org/10.1175/2011BAMS3158.1).
- Tian, Y., and Coauthors, 2009: Component analysis of errors in satellite-based precipitation estimates. *J. Geophys. Res.*, **114**, D24101, doi:[10.1029/2009JD011949](https://doi.org/10.1029/2009JD011949).
- Turk, F. J., and S. D. Miller, 2005: Toward improved characterization of remotely sensed precipitation regimes with MODIS/AMSR-E blended data techniques. *IEEE T Geosci. Remote Sens.*, **43**, 1059–1069, doi:[10.1109/TGRS.2004.841627](https://doi.org/10.1109/TGRS.2004.841627).
- Vincent, P., H. Larochelle, Y. Bengio, and P.-A. Manzagol, 2008: Extracting and composing robust features with denoising autoencoders. *Proc. 25th Int. Conf. on Machine Learning*, Helsinki, Finland, ACM, 1096–1103, doi:[10.1145/1390156.1390294](https://doi.org/10.1145/1390156.1390294).
- , —, I. Lajoie, Y. Bengio, and P.-A. Manzagol, 2010: Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, **11**, 3371–3408. [Available online at <http://www.jmlr.org/papers/volume11/vincent10a/vincent10a.pdf>.]
- Xie, J., L. Xu, and E. Chen, 2012: Image denoising and inpainting with deep neural networks. *Advances in Neural Information Processing Systems 25*, P. L. Bartlett et al., Eds., 350–358. [Available online at <http://papers.nips.cc/paper/4686-image-denoising-and-inpainting-with-deep-neural-networks.pdf>.]
- Zhang, C., B. Shahbaba, and H. Zhao, 2015: Hamiltonian Monte Carlo acceleration using neural network surrogate functions. ArXiv, accessed 28 January 2016. [Available online at <http://arxiv.org/abs/1506.05555>.]
- Zhou, G., K. Sohn, and H. Lee, 2012: Online incremental feature learning with denoising autoencoders. *J. Mach. Learn. Res.*, **22**, 1453–1461. [Available online at <http://jmlr.csail.mit.edu/proceedings/papers/v22/zhou12b/zhou12b.pdf>.]